Zadaća 3

Sara Ramadanović

Faculty of Electrical Engineering Sarajevo

Ovaj dokument je pripremljen u svrhu objašnjenja trećeg domaćeg zadatka za predmet Mašinsko učenje. Glavni zadaci ovog zadatka uključuju obradu i pripremu skupa podataka za buduću upotrebu. Metode koje su korištene tokom ovog zadatka su PAM k-medoidi i k-means algoritmi.

I. PRIPREMA PODATAKA

Set podataka koji je izabran za zadaću 3 je :(https://www.kaggle.com/ramamet4/app-store-apple-data-set-10k-apps?select=AppleStore.csv) dataset pod nazivom "*AppleStore.csv"*. Ovaj set podataka sadrži statističke podatke o ukupno 7000 aplikacija koje su trenutno objavljene u App store-u. Iako sam prvenstveno našla potpuni set podataka od ukupno 2.000.000 aplikacija, izabrala sam skraćenu verziju. Izvorni set podataka sadrži tačno 7197 instanci razvrstanih koje opisuje ukupno 17 atributa.

Iz izvornog seta podataka izbačene su kolone X, id i currency. Kolona X je izbačena jer predstavlja redni broj reda, kolona id izbačena je jer sadrži unikatni broj za određenu aplikaciju i iz tog razloga ove dvije kolone nemaju nikakve svrhe za narednu obradu podataka i smisla za dataset. Kolona currency je izbačena jer je prisutna samo jedna valuta a to je USD i svaka instanca u datasetu ima istu valutu(USD).

Set podataka je mješovitog tipa i sadrži kategoričke i numeričke varijable.

Kategoričke varijeble:

app_name
latest_version
age_rating
genre

Numeričke varijable:
size_bytes
price
all_ratings
current_ver__ratings
avg_user_rating
current_user_rating
num_devices_sup
num_screenshot
num_languages

Pored toga, izvršena je osnovna analiza podataka ista kao što smo obrađivali na zadaći 1 i 2, osim nekih grafičkih prikaza koji su izostavljeni radi skraćivanja same obrade podataka. Vrlo bitne analize koje sam izvršila su provjera NA vrijednosti i outliera. Ustanovljeno je da ovaj set podataka nema NA vrijednosti, tako da taj korak nije bilo potrebno odraditi.

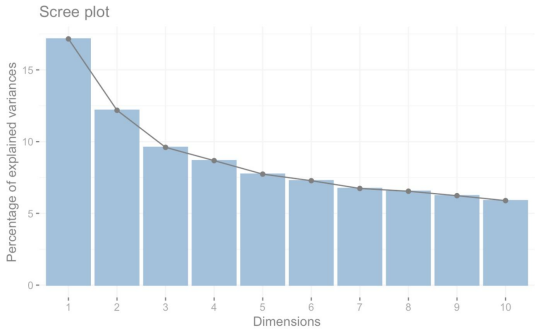
Tokom analize i provjere outliera, moglo se primijetiti da ovaj set podataka ima dosta nebalansiranih varijabli jer sadrže dosta outliera i vrijednosti koje odskaču od ostalih. Jedan primjer je atribut *all_ratings* koji sadrži podatke o ukupnom broju recenzija za određenu aplikaciju. Ti podaci znatno variraju jer imamo zastupljene vrlo popularne aplikacije koje imaju jako veliki broj recenzija, a neke druge manje popularne imaju znatno manji broj. Nakon nekoliko obrada transformacija i normalizacija podataka, dobila sam znatno lošije rezultate. S obzirom da je ovaj set podataka imao prilično dobre statističke vrijednosti, odlučila sam nastaviti sa obradom ovog seta.

II. PROTOTIP- BAZIRANI KLASTERING

Priprema podataka za klastering se sastojala iz učitavanja izlaznog dataseta nakon procesiranja. Prvenstveno je bilo potrebno prebaciti sve kategoričke varijable u numeričke jer PCA algoritam radi samo sa numeričkim podacima.

Nakon toga kreiran je PCA model uz pomoć prcomp() funkcije.

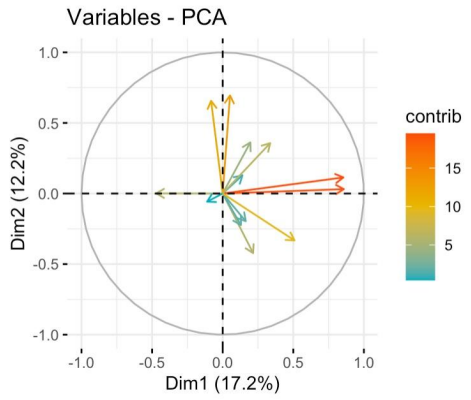
Scree plot:



rezultat summary() funkcije:

```
Importance of components:
                          PC1    PC2    PC3    PC4    PC5    PC6    PC7    PC8    PC9
PC10
Standard deviation     1.4935 1.2586 1.11720 1.06209 1.00283 0.97275 0.93592 0.92241 0.90025
0.87515
Proportion of Variance 0.1716 0.1218 0.09601 0.08677 0.07736 0.07279 0.06738 0.06545 0.06234
0.05891
Cumulative Proportion  0.1716 0.2934 0.38945 0.47622 0.55358 0.62637 0.69375 0.75920 0.82154
0.88045
                          PC11    PC12    PC13
Standard deviation     0.83636 0.79927 0.4645
Proportion of Variance 0.05381 0.04914 0.0166
Cumulative Proportion  0.93426 0.98340 1.0000
```

Summary() funkcija nam pokazuje značajnost komponenti i na osnovu slike iznad možemo vidjeti da PC1-PC11 imaju prag veći od 0.8, što je jako dobro.



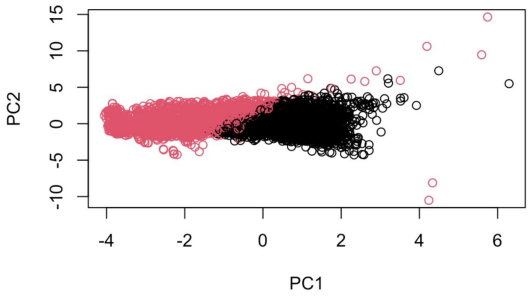
Nakon analize značajnosti atributa, odlučila sam zadržati prvih 10 atributa za daljnju analizu i onda je bilo potrebno izgraditi model uz pomoć funkcije predict() i tada je sve spremno za primjenu algoritama klasteringa.

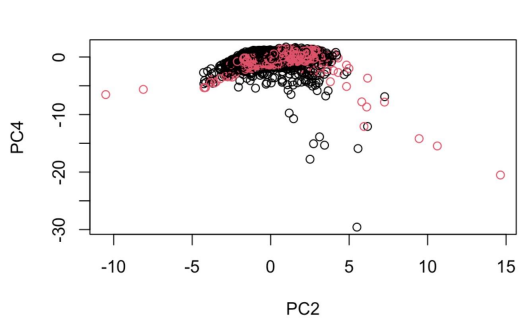
Uporede radi, prvo je vizuelno prikazan pam algoritam nad izvornim podacima i dao je sljedeće rezultate:

Vizualizacija je izvršena pomoću fviz_cluster() funkcije.

A. *PAM k-medoid* algoritam

Prvi algoritam koji je korišten jeste PAM k-medoid algoritam uz pomoć funkcije pam(), a vizualizacija klastera korištena je uz pomoć fviz_cluster().



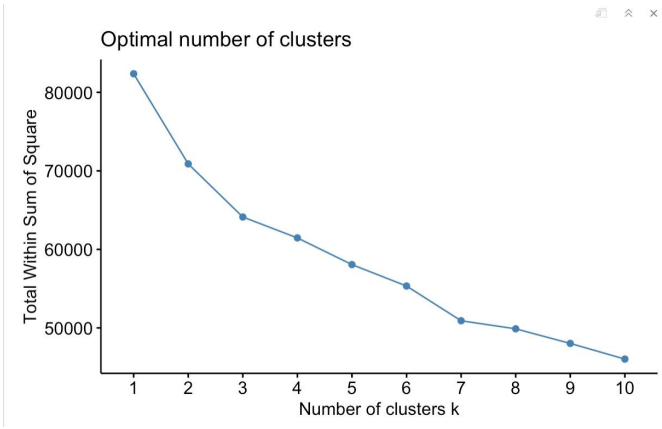


Kreirana su 2 klastera i na osnovu 3.slike vidimo dosta preklapanja između njih.

ODREĐIVANJE OPTIMALNOG BROJA KLASTERA

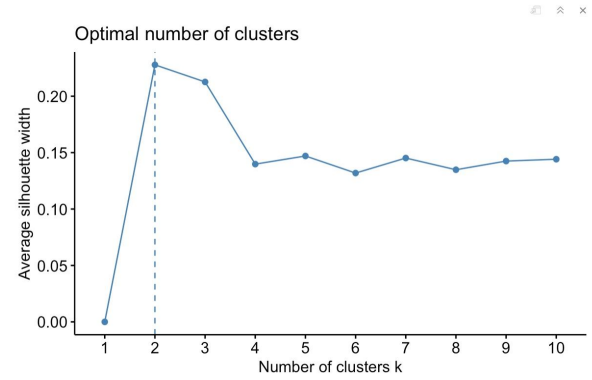
Sljedeći korak je određivanje optimalnog broja klastera za PAM algoritam i korištene su 3 metode: elbow, silhouette i gap.

elbow:



Optimalan broj klastera k = 7

silhouette:



Optimalan broj klastera k=2

gap:

Potrebno je izvršiti i procjenu da li dataset ima klastering tendencije. Iako nam je poznata metoda Hopkins statistike, nećemo je koristiti jer nije optimalno za ovaj slučaj i na osnovu hopkins statistike je dobiven rezultat da nema klastering tendencije nad ovim setom podataka.

Nasuprot tome, korištena je metoda računanja matrice distance.

Korištene su Manhattan, Euklidska, Minkowski i Pearson metrike. Izračunavanje ovih metrika u R studiu je bilo vrlo teško na mom računaru i iz tog razloga ove metrike izvršila sam samo nad PAM algoritmom.

Manhattan:

Euklid:

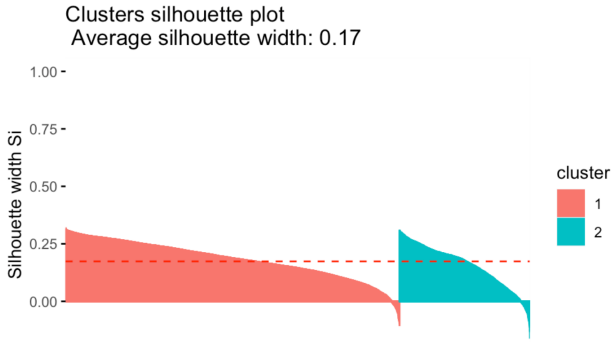
Minkowski:

VALIDACIJA KLASTERINGA

Pearson

Na osnovu grafičkih prikaza možemo vidjeti da su rezultati prilično slični, jedino se rezultat Pearson metrike malo više razlikuje od ostalih metrika.

Silhouette koeficijent:



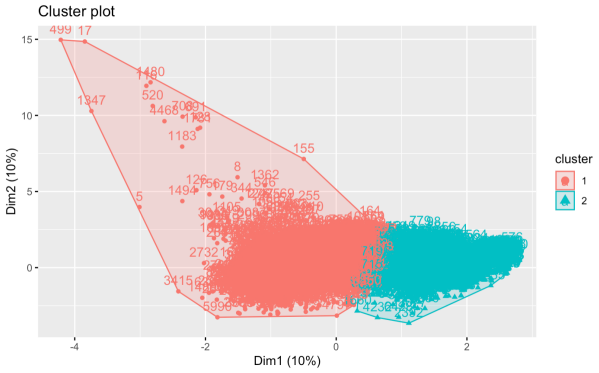

```
Srednja vrijednost silhouette metrike za PAM (manhattan metrika): 0.1737196
```

Zadnje na redu dolazi analiza čistoće klasteringa, međutim za to je potrebna labela klase koju ovaj set podataka ne posjeduje.

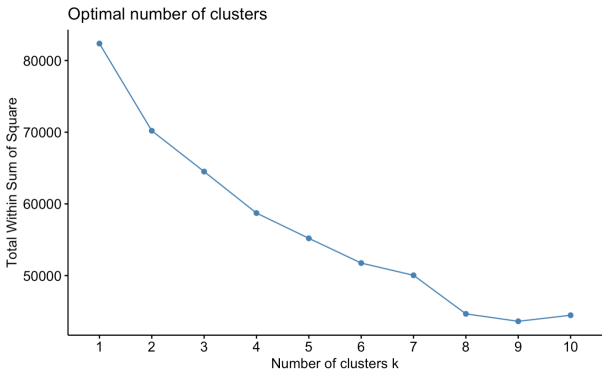
B. K-means algoritam

K- means algoritam je drugi algoritam koji je obrađen u sklopu analize podataka.

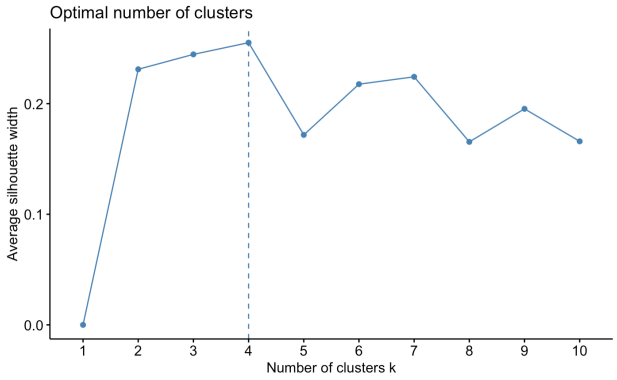
Grafički prikaz k means



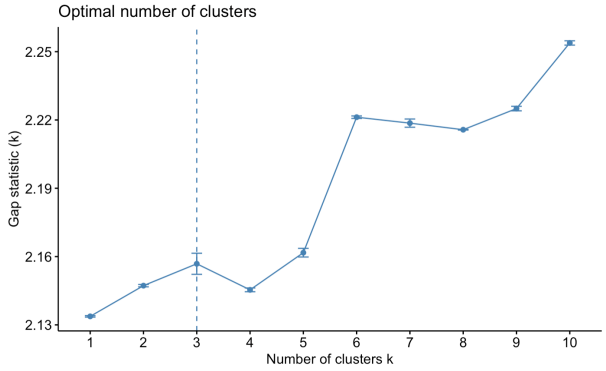
elbow:



silhouette:



gap:



Metrike distance za k-means nisu prikazane zbog poteškoća sa R studiom za vrijeme prikazivanja metrika za pam algoritam.